

IN THE CLAIMS:

Please amend the claims as indicated in the complete listing of pending claims listed below.

1. (Currently Amended) A data structure comprising:
~~an object, wherein the object includes a pointer to a vtable a table for virtual method dispatch and type identification~~, wherein the ~~vtable-table~~ includes a plurality of pointers, wherein the plurality of pointers point to a plurality of classes and wherein the plurality of classes include at least one unified type hierarchy to store a plurality of names from a plurality of programming languages for one implementation.
2. (Currently Amended) The data structure of claim 1 wherein the unified type hierarchy includes:
a first name from a first programming language, wherein the first name identifies an assigned object in the first programming language;
a second name from a second programming language, wherein the second name identifies the assigned object in the second programming language; and
a pointer to ~~a implementation~~an implementation of the assigned object.
3. (Original) The data structure of claim 2 wherein if the assigned object is not identified in the first programming language, then the first function name is a null.
4. (Original) The data structure of claim 1 wherein the unified type hierarchy includes:

a data structure that is recognizable by a first programming language and a second programming language.

5. (Original) The data structure of claim 1 wherein the data structure is a data structure for use in two or more hierarchical programming languages.
6. (Original) The data structure of claim 1 wherein the data structure is a data structure for use in two or more object-oriented programming languages.
7. (Original) The data structure of claim 6 wherein the two or more object-oriented programming languages include at least two of a group consisting of:
Java, C# (C Sharp), C++, Smalltalk, and Eiffel.
8. (Original) The data structure of claim 1 further comprising:
a root identifying each one of a plurality of programming languages wherein the data structure is recognizable in each one of the plurality of programming languages.
9. (Original) A method of identifying equivalent data structures comprising:
receiving a plurality of data structures, wherein the each one of the plurality of data structures are from a different one of a plurality of programming languages;
comparing the implementation of each one of the plurality of data structures; and
identifying at least two of the plurality of data structures that have identical implementations.

10. (Original) The method of claim 9 wherein the plurality of programming languages include at least two of a group consisting of Java, C# (C Sharp), C++, Smalltalk, and Eiffel.
11. (Original) A computer system comprising:
 - a processor;
 - an input/output system coupled to the processor via a bus system;
 - a memory system coupled to the bus, wherein the memory system includes processor executable instructions that when executed configure the processor to:
 - receive a plurality of data structures, wherein the each one of the plurality of data structures are from a different one of the plurality of programming languages;
 - compare the implementation of each one of the plurality of data structures;
 - and
 - identify at least two of the plurality of data structures that have identical implementations.
12. (Original) The system of claim 11, further comprising a network adapter coupled to the bus system and wherein the network adapter is coupled to a computer network.
13. (Original) The system of claim 11 wherein the plurality of programming languages include at least two of a group consisting of Java, C# (C Sharp), C++, Smalltalk, and Eiffel.
14. (Original) A method of unifying data structures comprising:

receiving a plurality of data structures, wherein the each one of the plurality of data structures are from a different one of a plurality of programming languages; comparing the implementation of each one of the plurality of data structures; identifying at least two of the plurality of data structures that have identical implementations; and

creating a unified data structure wherein the unified data structure includes:

a single implementation of the identified at least two data structures; and

a plurality of names of the identified at least two data structures.

15. (Original) The method of claim 14 wherein the plurality of names of the identified at least two data structures includes names that correspond to the respective one of the plurality of programming languages for each of the at least two data structures.
16. (Currently Amended) A computer system comprising:
 - a processor;
 - an input/output system coupled to the processor via a bus system;
 - a memory system coupled to the bus, wherein the memory system includes processor executable instructions that when executed configure the processor to:

~~A method of unifying equivalent data structures comprising:~~

receive a plurality of data structures, wherein the each one of the plurality of data structures are from a different one of a plurality of programming languages;

compare the implementation of each one of the plurality of data structures;

identify at least two of the plurality of data structures that have identical implementations; and

create a unified data structure wherein the unified data structure includes:
a single implementation of the identified at least two data structures;
and
a plurality of names of the identified at least two data structures.

17. (Original) The system of claim 16 wherein the plurality of programming languages include at least two of a group consisting of Java, C# (C Sharp), C++, Smalltalk, and Eiffel.
18. (Original) The system of claim 16 wherein the plurality of names of the identified at least two data structures includes names that correspond to the respective one of the plurality of programming languages for each of the at least two data structures.
19. (New) The method of claim 9, further comprising:
eliminating one of the identical implementations to create a unified data structure recognizable by the plurality of programming languages for a table for virtual method dispatch and type identification.
20. (New) The system of claim 11, wherein the memory system further includes processor executable instructions that when executed configure the processor to: create a unified data structure to contain names of the at least two of the plurality of data structures for the corresponding ones of the plurality of programming languages.